04AE-149

A Systematic Approach to Safety Case Management

Dr Tim Kelly
University of York, UK

ABSTRACT

In Europe, over recent years, there has been a marked shift in the regulatory approach to ensuring system safety. Whereas compliance with prescriptive safety codes and standards was previously the norm, the responsibility has now shifted back onto the developers and operators to construct and present well reasoned arguments that their systems achieve acceptable levels of safety. These arguments (together with supporting evidence) are typically referred to as a "safety case". This paper describes the role and purpose of a safety case (as defined by current safety and regulatory standards). Safety arguments within safety cases are often poorly communicated. This paper presents a technique called GSN (Goal Structuring Notation) that is increasingly being used in safety-critical industries to improve the structure, rigor, and clarity of safety arguments. Based upon the GSN approach, the paper also describes how an evolutionary and systematic approach to safety case construction, in step with system development, can be facilitated.

INTRODUCTION

A number of serious accidents such as the Piper Alpha Off-shore Oil and Gas Platform Disaster [1] and Clapham Rail Disaster [2] have been instrumental in prompting a reconsideration of how safety is managed in the safety-critical sector. In each of these cases, there had not been a total ignorance of safety concerns, or even a complete absence of safety standards. Instead, the underlying problem was that the designers and operators had failed to demonstrate a systematic and thorough consideration of safety. Safety standards introduced following these accidents (such as [3] and [4]) indicate a step change in the approach being adopted to safety regulation. Previous approaches have focussed primarily on prescriptive safety requirements, e.g. construction codes as described in [5]. With such approaches, operators claim safety through satisfaction of the *regulator's* requirements. With the introduction of *safety cases*, the responsibility is shifted back to the operators. It is up to the operators to demonstrate that they have an adequate argument of safety.

Despite the wide requirements for safety cases across many industries, it has been far from clear what constitutes a 'good' safety case, or how to construct a safety case. This is the subject of this paper.

DEFINING THE SAFETY CASE CONCEPT

The purpose of a safety case can be defined in the following terms:

> *A safety case should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context*

The concept of the 'safety case' has already been adopted across many industries (including defence, aerospace, and railways). Studying the safety standards relating to these sectors, it is possible to identify a number of definitions of the safety case – some clearer than others. The definition given above attempts to cleanly define the *core* concept that is in agreement with the majority of the definitions we have discovered.

The following are important aspects of the above definition:

- '**argument'** – Above all, the safety case exists to communicate an *argument*. It is used to *demonstrate* how someone can reasonably conclude that a system is acceptably safe from the evidence available.

- '**clear'** – A safety case is a device for *communicating* ideas and information, usually to a third party (e.g. a regulator). In order to do this convincingly, it must be as clear as possible.

- '**system'** – The system to which a safety case refers can be anything from a network of pipes or a software configuration to a set of operating procedures. The concept is not limited to consideration of conventional engineering 'design'.

- '**acceptably'** – Absolute safety is an unobtainable goal. Safety cases are there to convince someone

that the system is safe *enough* (when compared against some definition or notion of tolerable risk).

- '**context'** – Context-free safety is impossible to argue. Almost any system can be *unsafe* if used in an inappropriate or unexpected manner. (Consider arguing the safety of a conventional house-brick.) It is part of the job of the safety case to define the context within which safety is to be argued.

To elaborate the concept further, it is worth examining some alternative definitions briefly. The following definition is taken from the U.K. Ministry of Defence Ship Safety Management System Handbook JSP 430 [6].

> *"A safety case is a comprehensive and structured set of safety documentation which is aimed to ensure that the safety of a specific vessel or equipment can be demonstrated by reference to:*
> - *safety arrangements and organisation*
> - *safety analyses*
> - *compliance with the standards and best practice*
> - *acceptance tests*
> - *audits*
> - *inspections*
> - *feedback*
> - *provision made for safe use including emergency arrangements"*

This definition highlights two important aspects of the safety case. Firstly, it is a document. Some standards distinguish between the safety case as a *logical concept* (i.e. where the question, 'Does this system have a safety case?' is equivalent to asking 'Is this system acceptably safe?') and the safety case as a *physical artefact* (sometimes called the Safety Case *Report*). As is commonly done, this definition uses the term safety case synonymously with the documentation that presents the safety case. Secondly, it makes clear that the nature of the safety case is to refer to, and pull together, potentially many other pieces of information (such as safety analyses).

A more mechanistic definition of the *software* safety case is that used by the U.K. Ministry of Defence Standard 00-55 [7]. Although referring to software systems, it is not difficult to see how such a definition translates to other systems.

> *"The software safety case shall present a well-organised and reasoned justification based on objective evidence, that the software does or will satisfy the safety aspects of the Statement of Technical Requirements and the Software Requirements Specification."*

This definition makes clear the role of the safety case in expressing satisfaction of specific *Safety Requirements* or *Objectives.* It is rare that acceptable safety is a completely undefined concept. Within industry sectors, and for particular classes of system, definitions of acceptable safety have evolved. These may be expressed in terms of prescriptive requirements, development codes or assessment principles. For example, Defence Standard 00-55 expresses many individual requirements concerning the development and assessment of safety critical software systems. Prescriptive requirements are a third party expression of a high-level safety argument – where meeting requirements implies some degree of safety. The safety case must clearly identify and address applicable requirements.

REQUIREMENTS, ARGUMENT AND EVIDENCE

Underlying the descriptions of the safety case given in the previous section is a view of the safety case consisting of three principal elements: Requirements, Argument and Evidence. The relationship between these three elements is depicted in Figure 1.

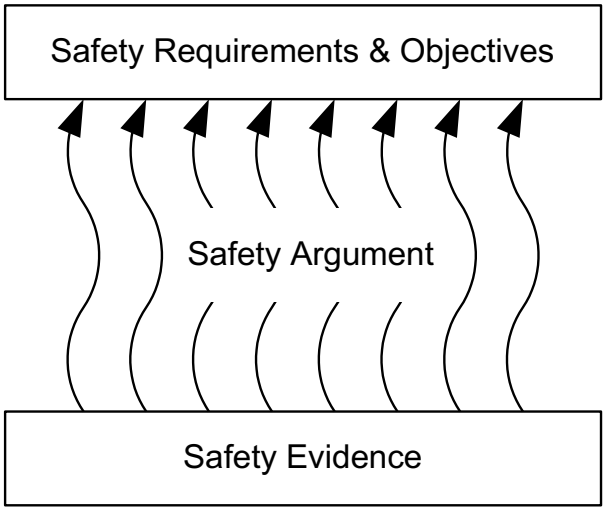


Figure 1 – The Role of Safety Argumentation

The safety argument is that which communicates the relationship between the evidence and objectives. Based on the author's personal experience, gained from reviewing a number of safety cases, and validated through discussion with many safety practitioners (some directly responsible for reviewing and accepting safety cases), a commonly observed failing of safety cases is that the *role of the safety argument is often neglected*. In such safety cases, many pages of supporting evidence are often presented (e.g. hundreds of pages of fault trees or Failure Modes and Effects Analysis tables), but little is done to explain how this evidence relates to the safety objectives. The reader is often left to guess at an unwritten and implicit argument.

Both argument and evidence are crucial elements of the safety case that must go hand-in-hand. Argument without supporting evidence is unfounded, and therefore unconvincing. Evidence without argument is unexplained – it can be unclear that (or how) safety objectives have been satisfied. In the following section we examine how safety arguments may be clearly communicated within safety case reports.

SAFETY CASE REPORTS AND SAFETY ARGUMENTS

Conventionally the safety case is thought of as a report. Many safety standards (such as Defence Standard 00-55 [7]) even go so far as to define the expected structure and contents of safety case reports. Whilst there are variations between the recommendations of the standards the following list illustrates the most typical headings expected within a safety case report:

- Scope
- System Description
- System Hazards
- Safety Requirements
- Risk Assessment
- Hazard Control / Risk Reduction Measures
- Safety Analysis / Test
- Safety Management System
- Development Process Justification
- Conclusions

The 'Scope' section plays an important role within the safety case. As mentioned earlier, a safety case cannot argue the safety of a system in *any* context. There have to be clearly defined limits as to the scope of applicability of the safety argument being presented within the safety case. Specific inclusions and exclusions (such as sabotage) must be identified. The scope must identify the expected lifetime and duration of the system (and corresponding argument). The 'Scope' section therefore defines the context within which the remainder of the safety argument sits.

It is usual to present an overview of the system within the safety case. This is the role of the 'System Description' section. It is not the purpose of this section to provide *full* design detail. For this, we would expect the safety case to refer to the original design documentation. However, *some* description is typically necessary in order for the reader of the safety case document to understand the sections that follow. For example, sufficient system description is necessary in order to make sense of the system hazards and safety requirements when described later in the document. Importantly, the system description section needs to make clear the version of the system being discussed. Additionally, when talking about a system that forms part of a larger configuration of systems, this section must make clear the boundaries and interfaces included within the scope of the safety argument.

The purpose of the 'Safety Hazards' and 'Safety Requirements' is straightforward. In the section on hazards the safety case must describe the key hazards posed by the system in question. The primary container of information relating to hazards should remain the *Hazard Log* (e.g. as defined by Defence Standard 00-56 [8]). The purpose of this section is simply to *summarise* the identified hazards. All safety requirements should similarly be brought together and summarised in the 'Safety Requirements' section. Safety requirements may arise from a wide range of sources – the customer, safety standards, derived from hazard analysis, and/or requirements cascading down higher-level systems.

The purpose of the 'Risk Assessment' section is to describe the assessed level of residual risk associated with each of the identified hazards. The residual risk is the risk remaining after the risk reduction measures (described in the next section of the safety case report) have been applied. This section would typically discuss how the assessed level of risk compares with established risk acceptance criteria (e.g. tolerable probabilities for given severity events – as practiced in the aerospace arena [9]). If appropriate, ALARP (As Low as Reasonably Practicable) arguments may also be presented within this section.

The following two sections ('Risk Reduction Measures' and 'Safety Analysis') present the 'heart' of the product safety argument. The first of these sections presents the technical discussion of the risk reduction measures (whether reduction in probability of hazard occurrence or mitigation *of* hazard occurrence) that have been deployed within the system design. The argument (even if only implicitly) is that these measures are sufficient. This argument must be backed by the following section ('Safety Analysis') that presents an overview of the safety evidence available (analysis, test, inspection, in-service evidence etc.) and how it *justifies the adequacy* and *sufficiency* of the measures adopted. As with the system description section, it is important to note that the 'Safety Analysis' section presents *only* a summary of the evidence available – it is typical for the evidence (e.g. test results) to maintained in separate reports and for the safety case to merely *refer* to them.

The penultimate two sections ('Safety Management System' and 'Development Process Justification') present *process* safety arguments. From such arguments confidence in the safety of the system is built upon knowledge of the design and safety *processes* adopted during the development and assessment of the system. Process safety arguments are typically regarded as weaker than product arguments [10]. Nevertheless, they are widely used to build confidence and can add value when used alongside a 'direct'

product argument. The first of these sections 'closes the loop' on the safety management practice planned in the System Safety Programme Plan (an early lifecycle planning artefact). In the safety plan audits, reviews, appointment of independent assessors and other key safety personnel (e.g. the Project Safety Committee) will have been planned. The role of this section of the safety case is to present the argument and evidence that the plan was carried out effectively. The latter of these two sections ('Development Process Justification') presents the arguments most typically associated with System Integrity Level (SIL) justification – namely, that the tools, techniques and methods adopted on the project were appropriate given the level of safety risks involved. Such arguments may, for example, justify the adoption of a specific programming language and testing regime for a software based system.

Finally, the safety case report should present the key conclusions and high level findings that convince the reader that the system is acceptably safe to operate in its intended design context.

Whilst the report-oriented view presented above is helpful, when adopting such a view it is often possible to lose sight of the *logical chain of reasoning* (the safety argument) that should be running through the safety case. Creation of a document with the headings described above is insufficient to establish a safety case. Indeed, it is possible to possess a document *called* the Safety Case and for there to be no safety case (i.e. there is no compelling safety argument). In the next section we describe how safety arguments are typically communicated within any safety case report.

COMMUNICATING SAFETY ARGUMENTS

Safety arguments are most typically communicated in existing safety cases through free text. Figure 2 shows a fragment of a safety argument communicated using free text.

> The Defence in Depth principle (P65) has been addressed in this system through the provision of the following:
>
> - Multiple physical barriers between hazard source and the environment (see Section X)
> - A protection system to prevent breach of these barriers and to mitigate the effects of a barrier being breached (see Section Y)

Figure 2 – An Example Textual Safety Argument

In Figure 2, the text describes clearly how a safety requirement (P65) has been interpreted and achieved in the system. It also clearly provides references to where the evidence supporting the lower level statements can be found.

Well-structured approaches to expressing safety arguments in text can be effective (as shown in Figure). However, there are problems experienced when text is the only medium available for expressing complex arguments. The text shown in Figure 3, taken from a real industrial safety case (with identification of the target application hidden), illustrates some of these problems.

> For hazards associated with warnings, the assumptions of [7] Section 3.4 associated with the requirement to present a warning when no equipment failure has occurred are carried forward. In particular, with respect to hazard 17 in section 5.7 [4] that for test operation, operating limits will need to be introduced to protect against the hazard, whilst further data is gathered to determine the extent of the problem.

Figure 3 – The Problems of Textual Safety Arguments

The underlying problem of the text shown in Figure 3 is that it is unclear and poorly structured English. Not all engineers responsible for producing safety cases write clear and well-structured English. Consequently, the meaning of the text, and therefore the structure of the safety argument, can be ambiguous and unclear. Cross-references, of the type shown in Figure 3, are often necessary given the role of the safety case as an integrator of evidence. However, multiple cross-references in text can be awkward and can disrupt the flow of the main argument.

In the context of developing, agreeing, and maintaining the safety arguments within the safety case, the biggest problem with the use of free text is in ensuring that all stakeholders involved share the same understanding of the argument. Without a clear and shared understanding of the argument, safety case management is often an inefficient and ill-defined activity.

The following section describes a structured technique that has been developed to address the problems of clearly expressing and presenting safety arguments.

THE GOAL STRUCTURING NOTATION

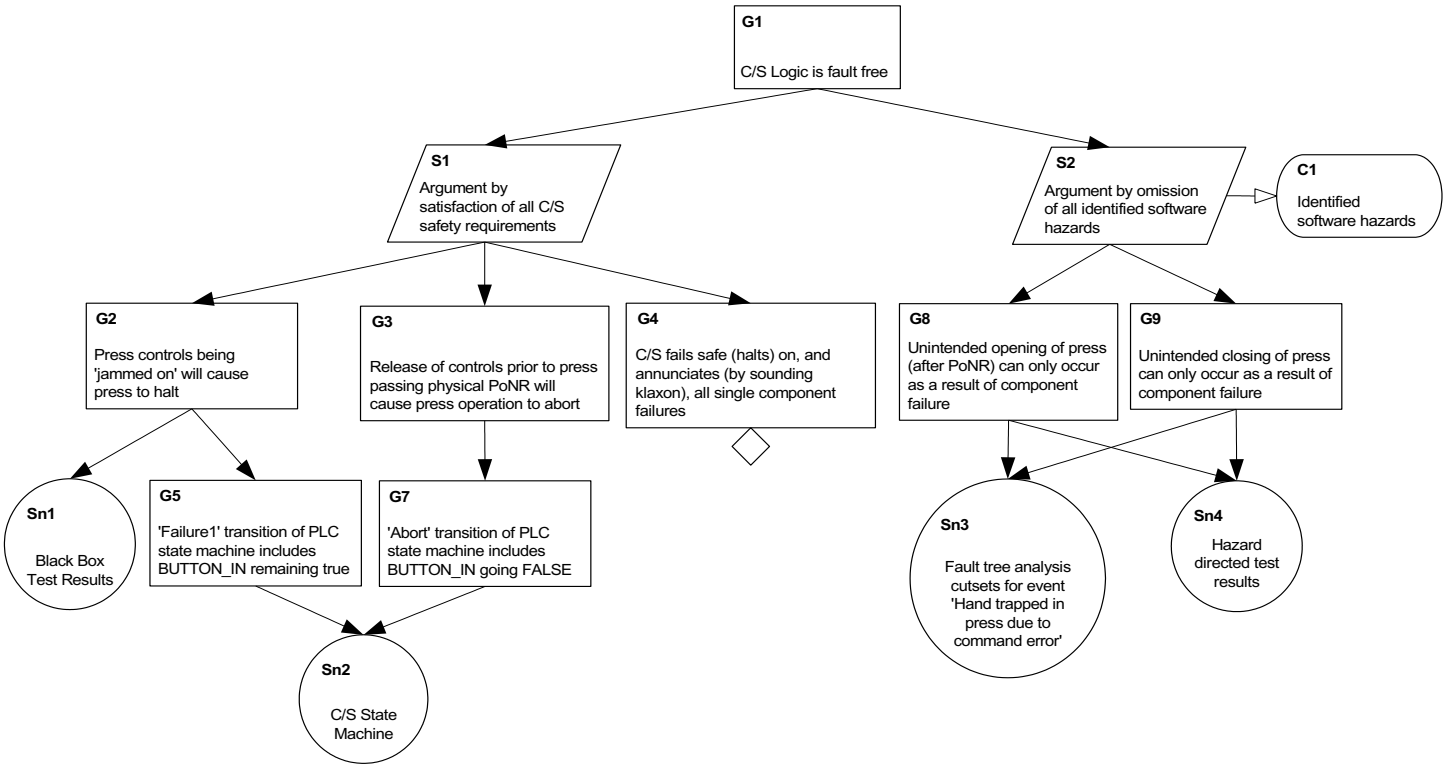


Figure 5 – An Example Goal Structure

The Goal Structuring Notation (GSN) [11] - a graphical argumentation notation - explicitly represents the individual elements of any safety argument (requirements, claims, evidence and context) and (perhaps more significantly) the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument). The principal symbols of the notation are shown in Figure 4 (with example instances of each concept).

When the elements of the GSN are linked together in a network they are described as a '*goal structure*'. The principal purpose of any goal structure is to show how goals (claims about the system) are successively broken down into sub-goals until a point is reached where claims can be supported by direct reference to available evidence (solutions). As part of this decomposition, using the GSN it is also possible to make clear the argument strategies adopted (e.g. adopting a quantitative or qualitative approach), the rationale for the approach and the context in which goals are stated (e.g. the system scope or the assumed operational role).

Figure 5 shows an example goal structure. In this structure, as in most, there exist 'top level' goals – statements that the goal structure is designed to support. In this case, "*C/S (Control System) Logic is fault free*", is the (singular) top level goal. Beneath the top level goal or goals, the structure is broken down into sub-goals, either directly or, as in this case, indirectly through a strategy. The two argument strategies put forward as a means of addressing the top level goal in Figure 5 *are "Argument by satisfaction of all C/S (Control System) safety requirements*", and, "*Argument by omission of all identified software hazards*". These strategies are then substantiated by five sub-goals. At some stage in a goal structure, a goal statement is put forward that need not be broken down and can be clearly supported by reference to some evidence. In this case, the goal *"Unintended Closing of press after PoNR (Point of No Return) can only occur as a result of component failure*", is supported by direct reference to the solutions, "*Fault tree cutsets …*" and "*Hazard Directed Testing Results*".

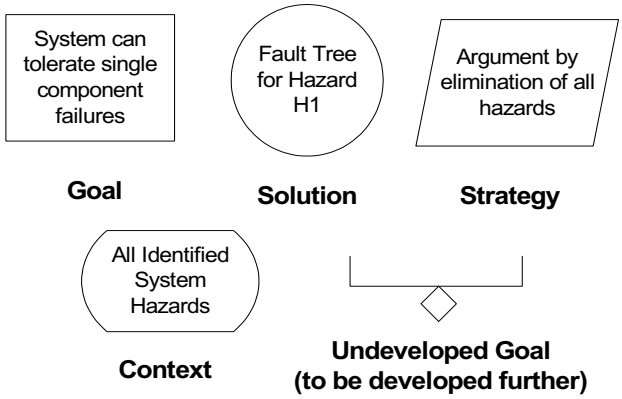


Figure 4- Principal Elements of the Goal Structuring Notation

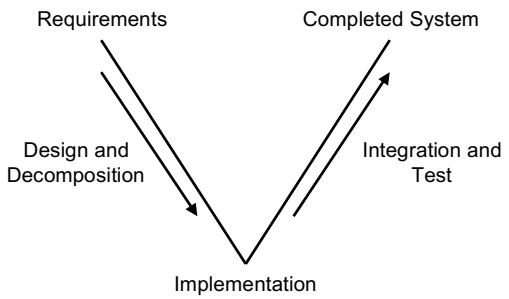
Within Europe, GSN has been adopted by a growing number of companies within safety-critical industries (such as aerospace, railways and defence) for the presentation of safety arguments within safety cases. The following list includes some of the applications of GSN to date:

- Eurofighter Aircraft Avionics Safety Justification
- Hawk Aircraft Safety Justification
- U.K. Ministry of Defence Site Safety Justifications
- U.K. Dorset Coast Railway Re-signalling Safety Justification
- Submarine Propulsion Safety Justifications
- Safety Justification of UK Military Air Traffic Management Systems
- London Underground Jubilee Line Extension Safety Justification
- Swedish Air Traffic Control Applications
- Rolls-Royce Trent Engine Control Systems Safety Arguments

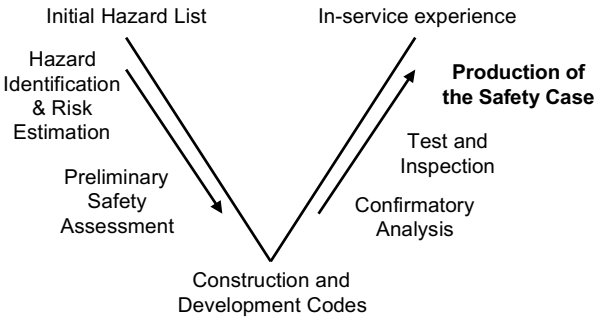
The key benefit experienced by those companies adopting GSN is that it improves the comprehension of the safety argument amongst all of the key project stakeholders (i.e. system developers, safety engineers, independent assessors and certification authorities). In turn, this has improved the quality of the debate and discussion amongst the stakeholders and has reduced the time taken to reach agreement on the argument approaches being adopted. However, having a clear means of communicating safety arguments is only a partial answer to the challenge of establishing a systematic safety case development approach. In addition, it is important to consider the *timing* of safety case development with respect to the system development lifecycle. The following section discusses this issue.

SAFETY CASE DEVELOPMENT LIFECYCLE

It is increasingly recognised by both safety case practitioners and many safety standards that safety case development, contrary to what may historically have been practised, cannot be left as an activity to be performed towards the end of the safety lifecycle. This view of safety case production being left until all analysis and development is completed is depicted in Figure 6.



Design Lifecycle



Safety Lifecycle

Figure 6 - A Historical View of Safety Case Development

A traditional view of the design and development lifecycle is shown in the upper half of Figure 6. Running concurrently with this, shown in the lower half of the diagram, is the historical view of the safety lifecycle, showing safety case development as a discrete activity to be performed following the completion of the safety assessment activities.

The problems that have been experienced with this style of safety case development include [12]:

- Large amounts of re-design resulting from a belated realisation that a satisfactory safety argument cannot be constructed. In extreme cases, this has resulted in 'finished' products having to be completely discarded and redeveloped.
- Less robust safety arguments being presented in the final safety case. Safety case developers are forced to argue over a design as it is given to them – rather than being able to influence the design in such a way as to improve safety and improve the nature of the safety argument. This can result in, for example, probabilistic arguments being relied upon more heavily than deterministic arguments based upon explicit design features (the latter being often more convincing).
- Lost safety rationale. The rationale concerning the safety aspects of the design is best recorded at 'design-time'. Where capture of the safety argument is left until after design and implementation – it is possible to lose some of the safety aspects of the design decision making process which, if available, could strengthen the final safety case.

Unfortunately, though not surprisingly, few practitioners are prepared to publicise failures of this style of safety case development. However, Cullen in [11] presents some of the experiences of BNFL in producing a safety case for the Sellafield Alpha Reduction Plant. For this plant he relates that a 'traditional' approach was first

adopted – where "plant design has proceeded more or less independently of the production of the safety case". Design progressed to the firm proposal stage before being passed to the Safety Department. Significant safety hazards were identified with this proposal – making it impossible to produce a convincing safety case. A re-design was therefore required – resulting in great expense. The re-design was again developed into a firm proposal before the safety case was considered. However, this time, other significant problems were found with the new proposal, requiring more (expensive) re-design. It was only on the third re-design, where consideration of the safety case was integrated into the design requirements that an acceptable, arguably safe, design resulted.

INCREMENTAL SAFETY CASE DEVELOPMENT

Safety standards, such as the U.K. Defence Standards 00-56 [8] and Ship Safety Management Handbook JSP430 [6] now require that safety case development be treated as an evolutionary activity that is integrated with the rest of the design and safety lifecycle. Defence Standard 00-56 states the following:

> *"The Safety Case should be initiated at the earliest possible stage in the Safety Programme so that hazards are identified and dealt with while the opportunities for their exclusion exist"*

Equally, JSP 430 requires the following:

> *"The Safety Case is to be prepared in outline at presentation of the Staff Requirement and is to be updated at each major procurement milestone up to and including hand-over from the procurement to the maintenance authority … Ideally there should be a seamless development of the Safety Case from one phase to the next"*

The interpretation of this 'seamless development' that is being adopted by the majority of the safety standards is the production and presentation of the safety case at a number of stages during the development of a project. For example, Defence Standard 00-55 [7] talks of formally issuing three versions of the (Software) Safety Case:

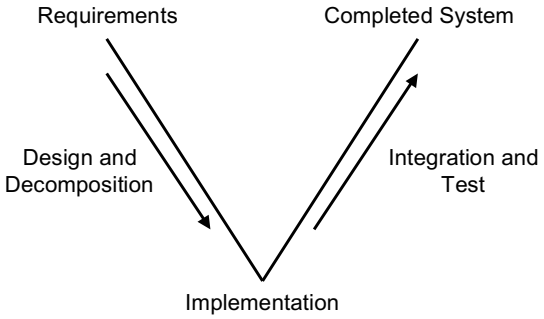
- **Preliminary Safety Case** – after definition and review of the system requirements specification
- **Interim Safety Case** – after initial system design and preliminary validation activities
- **Operational Safety Case** – just prior to in-service use, including complete evidence of satisfaction of systems requirements

The integration between the production of these safety cases and the traditional development lifecycle is depicted in Figure 7.

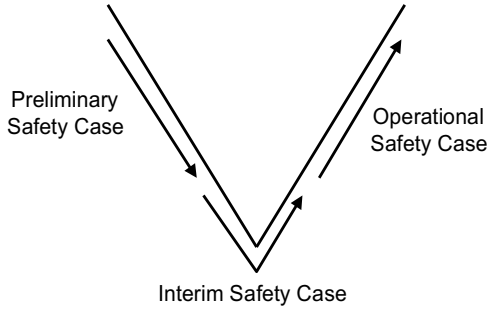
There is often some variation on the above requirements between regulatory domains. For example, for civil nuclear power generation in the UK safety cases are additionally required at certain milestones in the project. In the commissioning of Sizewell 'B' safety cases were presented prior to first fuel load, prior to first generation of power and prior to being allowed to export power to the national grid [13]. However, regardless of the specifics of numbers of safety cases and timings of submissions, the principle of phased safety case production is increasingly being accepted as a core concept across all domains.

EVOLVING SAFETY ARGUMENTS

At the heart of the concept of phased safety case production is the presentation of an *evolving safety argument*. At the Preliminary Safety Case stage the aim is to present an outline safety argument showing the principal objectives, approach to arguing safety and the forms of evidence anticipated. At the Interim stage the argument should be evolved to reflect the increased knowledge concerning the detailed design and specification of the system. At the Operational stage the argument can again be evolved further to reflect evidence concerning the system as implemented and tested.



Design Lifecycle



Safety Case Lifecycle

Figure 7 – An Integrated View of Safety Case Development

On any safety-critical / safety-related project it is crucial to gain an understanding of the form and expected content of the safety (or certification) argument as early as possible. As indicated by the quote from Defence Standard 00-56 provided in the previous section, early identification of safety objectives allows the design to be influenced as system development progresses in order that a more compelling safety case may be established. In the following section, we discuss in more detail the role and significance of establishing a *preliminary safety case*.

THE PRELIMINARY SAFETY CASE

The Preliminary Safety Case will typically be prepared in a project after the following activities have been performed:

- **Production of Safety Plan -** Definition of the key safety processes, roles and responsibilities to be enacted during system development.
- **Identification of Required Safety Properties -** Including identification of applicable safety standards, the requirements from these standards that apply to the system under development and customer-desired safety properties.
- **Preliminary Hazard Analysis (PHA) -** Identification of potential system hazards through systematic review of the initial, top-level, system design – e.g. for a software system, through performing Software Hazard and Operability Studies (HAZOPS) [14] over a high-level data flow diagram that defines the key processes and data flows.
- **Risk Estimation -** Estimation of the level of risk posed by each of the identified system hazards – e.g. through qualitative description of both the severity and likelihood of hazard consequences and use of a Hazard Risk Index (HRI) Matrix.
- **Identification of Failure Rate and Integrity Level Requirements -** Predominantly, identifying the principal requirements implied by the risks identified in the Risk Estimation exercise – e.g. tolerable failure rate targets for each risk category.

Notably, however, the Preliminary Safety Case will usually be prepared prior to there being any ***detailed*** *system design or specification*, and therefore before any detailed system safety analysis or testing is possible. Given the absence of design detail, one might question the value of producing a safety case at this stage in the project. However, the document can fulfil the following objectives:

- Defining the **scope** of consideration for the (final) safety case
- Declaring what have been identified as the **key safety issues and objectives** associated with the system - the principal System Hazards, Safety Requirements and Applicable Standards
- Defining the **approach** that is being adopted in arguing safety – including the key techniques and sources of **supporting evidence** to be employed
- Defining (safety-relevant) development **procedures** that will be enacted during system development, e.g. the languages, methods and tools to be used for each Software Integrity Level

Having fulfilled these objectives, submitting the Preliminary Safety Case to the customer (regulator) provides an early opportunity to get agreement, even if only informally, on the certification approach being adopted. In addition, the Preliminary Safety Case helps the developer to clearly set out the safety context within which the project must be executed. Through the document, safety objectives to be achieved are flagged in advance of system development – reducing the extent to which requirements will be 'discovered' after significant functional design commitments have already been made.

PRELIMINARY SAFETY ARGUMENTS

As with both the Interim and final Operational Safety Cases, the Preliminary Safety Case will typically present information under the following headings: (Under each heading we describe the contents that could be expected at the time of producing the Preliminary Safety Case.)

- **Scope -** Boundary of concern, standards to be addressed, relationship to other systems / extant safety cases
- **System Description** - High-Level (Preliminary) Overview of the System: Key functions + Outline of Physical Elements
- **System Hazards** - Results of Preliminary Hazard Analysis - Key Credible Hazards. (These may well change for later submissions of the safety case.)
- **Safety Requirements -** Description of Top-level safety requirements (emerging from study of the standards, and the Preliminary Hazard Analysis), e.g. Failure Rate in particular Failure Modes.
- **Risk Assessment** - Results of Risk Estimation exercise, Accident Sequences, HRI used and the resulting Risk Classes for all identified Hazards. (As with the System Hazards – the assigned Risk Classes may change for later submissions of the safety case.)
- **Hazard Control / Risk Reduction Measures -** At this stage - how the project plans to tackle each identified risk - design measures, protection systems, redundancy etc.
- **Safety Analysis / Test** - At this stage - how the project intends to provide evidence of successful deployment of risk reduction measures, meeting failure rate targets, demonstrating correctness, etc.
- **Safety Management System -** Reference to contents of Safety Plan for roles, responsibilities,

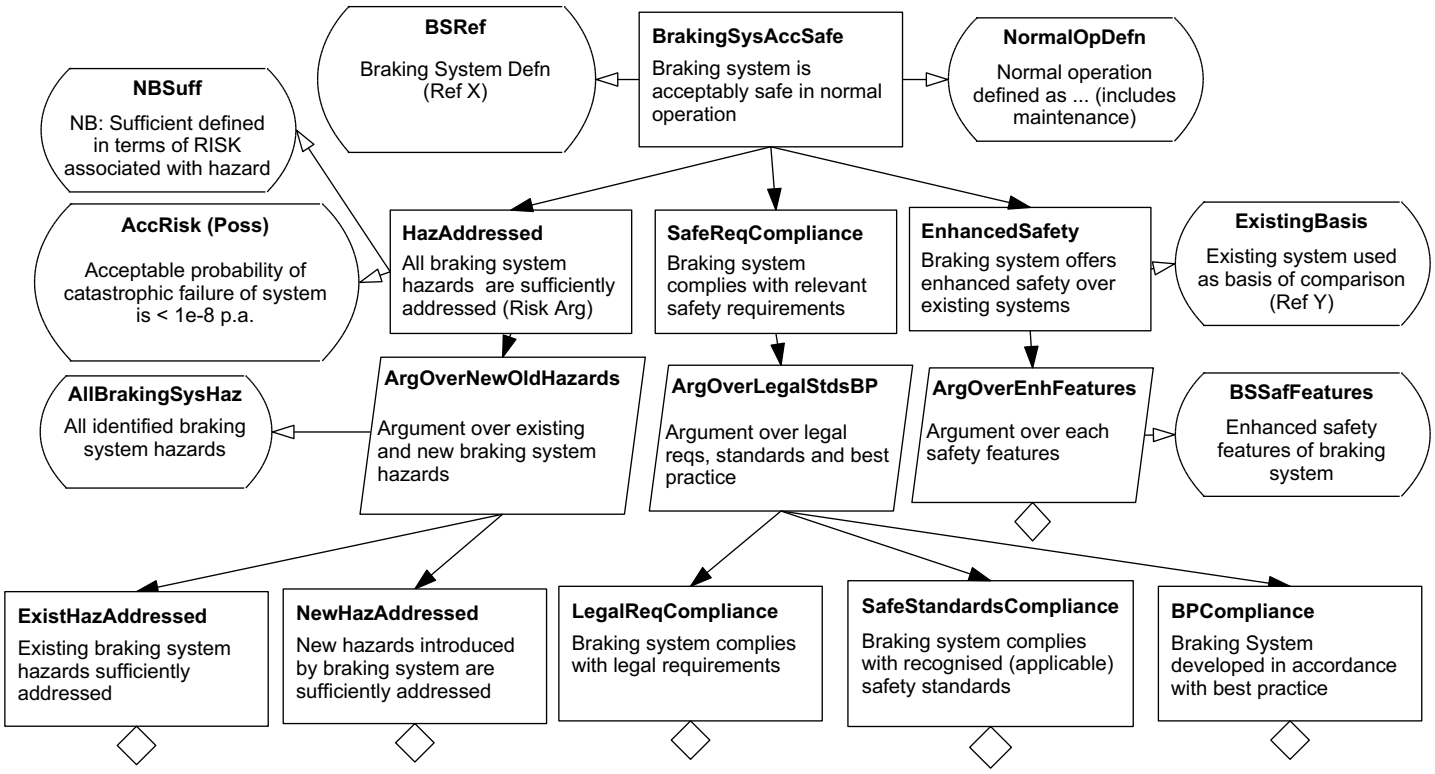


Figure 8 – The Beginnings of a Preliminary Safety Argument for a Braking System

procedures. (This will be fairly stable for later safety cases.)

- **Development Process Justification -** An outline of the development procedures, design methodologies to be used, coding standards, change control procedures etc. and how these will be shown to meet integrity level, or development assurance level, requirements.
- **Conclusions -** At this stage - the key reasons why the project believes that the system will be safe to deploy the system, what will be concluded from analysis and test evidence etc.

Although each of the elements listed above forms a necessary part of the Preliminary Safety Case, as stated earlier one of the principal objectives is to obtain general agreement with the customer as to *the argument approach* being adopted on the project. To do this, it can be useful to explicitly present a *Preliminary Safety Argument* that describes the emergent safety requirements, the interpretation of these requirements and points forward to the claims that will be made about the system and the evidence that will be used in support of these claims. GSN can provide a useful means of mapping out such arguments. Figure 8 illustrates the use of GSN to outline the preliminary (top-level) safety argument for a car braking system.

The beginnings of a preliminary safety argument presented in Figure 8 clearly outlines a three-pronged approach to arguing the safety of the (new) braking system. Firstly, a conventional hazard mitigation argument is presented. Secondly, a 'compliance with standards' argument is put forward. Thirdly, a comparative argument claiming improved safety over existing systems is proposed.

Importantly, the use of GSN has highlighted the need for context to be defined. At the top level, the need for a clear definition of the braking system and 'normal operation' of the system is signaled (the latter being highly significant in 'scoping' the supporting argument). For the hazard based argument, the highlighted context indicates that knowledge of the system hazards in addition to some overall risk criteria will be required. In the lower levels of the hazard argument the safety argument authors have chosen to split the argument into new and existing hazards (believing that different supporting arguments and evidence will be used between the two claims).

The standards compliance argument divides into three specific claims – regarding compliance with legal requirements, safety standards and then 'best practice'. (Context may well need to be added to this last at a later stage in order to define what constitutes best practice!)

Finally, the argument claiming an improvement on existing braking systems is structured according to the safety features offered by the new braking system (e.g. Emergency Brake Assist). Again, context is used to show that this argument can only be put forward when the baseline for comparison has been established.

Establishing a preliminary safety argument, such as the one shown in Figure 8, serves as a foundation from

which the safety case can evolve in step with system development. As the system development progresses, so should the safety argument. Adopting such an approach enables confidence in the feasibility of establishing an acceptable safety case to grow throughout the safety lifecycle – as the argument is better understood and the detail (including supporting evidence) added. In this way, the potential *project* risk associated with failure to ultimately gain system certification / 'safety acceptance' is being addressed.

CONCLUSION

In this paper we have described the safety case concept as adopted by many safety critical industries (such as defence, railways and aerospace) within Europe. The principal objective of a safety case is to present an argument that a system is acceptably safe to operate in a given context. However, the safety *argument* is often poorly communicated through the textual narrative of safety case reports. The Goal Structuring Notation (GSN), presented within this paper, has been developed to provide a clear, structured, approach to developing and presenting safety arguments. Alongside clear presentation, systematic safety case development relies upon timely consideration of the safety case in parallel with system development. This paper has also described the (now widely endorsed) approach of phased safety case development and, within this context, illustrated how GSN can serve in presenting evolving safety arguments.

REFERENCES

1. W. D. Cullen, "The Public Enquiry into the Piper Alpha Disaster," Department of Energy, London, HMSO November 1990.
2. C. Edwards, "Railway Safety Cases," presented at Safety and Reliability of Software Based Systems - Twelfth Annual CSR Workshop, Bruges, Belgium, 1997.
3. U.K. Health and Safety Executive, "A guide to the Offshore Installations (Safety Case) Regulations 1992," Health and Safety Executive, HSE Books 1992.
4. U.K. Health and Safety Executive, "Railway Safety Cases - Railway (Safety Case) Regulations 1994 - Guidance on Regulations," Health and Safety Executive, HSE Books 1994
5. R. Chuse, *Pressure vessels: the ASME code simplified*, 7th ed. New York; London: McGraw-Hill, 1993.
6. U.K. Ministry of Defence, "JSP 430 - Ship Safety Management System Handbook," Ministry of Defence January 1996.
7. U.K. Ministry of Defence, "00-55 Requirements of Safety Related Software in Defence Equipment," Ministry of Defence, Defence Standard August 1997.
8. U.K. Ministry of Defence, "00-56 Safety Management Requirements for Defence Systems," Ministry of Defence, Defence Standard December 1996.
9. Joint Aviation Authorities, "Joint Airworthiness Requirements JAR-25: Large Aeroplanes (Change 13)," Civil Aviation Authority October 1989.
10. R. A. Weaver, J. A. McDermid, T. P. Kelly, Software Safety Arguments: Towards a Systematic Categorisation of Evidence, Presented at the 20th International System Safety Conference (ISSC 2002), Denver, Colorado, USA, 2002, System Safety Society
11. T. P. Kelly, Arguing Safety – A Systematic Approach to Safety Case Management, DPhil Thesis YCST99-05, Department of Computer Science, University of York, UK, 1998
12. R. J. Cullen, "Safety as a Design Tool," presented at Managing Risk in a Changing Organisation Climate - Proceedings of the Safety and Reliability Symposium, Swindon, U.K., 1996.
13. IEE, *Proceedings of International Conference on Sizewell B: The First Cycle*. London: IEE, 1996.
14. U.K. Ministry of Defence, Defence Standard 00-58 HAZOP Studies on Systems Containing Programmable Electronics Issue 2, Ministry of Defence, 2000

CONTACT

Dr Tim Kelly is a Lecturer in critical systems engineering within the Department of Computer Science at the University of York. He is also Deputy Director of the Rolls-Royce Systems and Software Engineering University Technology Centre (UTC) at York. His expertise lies predominantly in the areas of safety case development and management. Tim has provided extensive consultative and facilitative support in the production of acceptable safety cases for companies from the medical, aerospace, railways and power generation sectors. Tim has a PhD from the University of York and a MA from the University of Cambridge.

E-mail: tim.kelly@cs.york.ac.uk

Web: http://www.cs.york.ac.uk/~tpk